The Tor Browser, Vulnerabilities and Solutions for the Common User

Lin, Katherine
NYU Tandon School of Engineering
NY, USA
klc616@nyu.edu

Abstract - Most Tor users are everyday people with little technical skills in IT. Because of the gaps and inaccuracies in the understanding of the operation and threat model of Tor, the use of the tool can lead to deanonymizations that may cause serious repercussions. Some of them may be account compromises, identity theft, financial losses (resulting from fraud), surveillance of communication and movements. Our findings mainly aim to educate and improve the way people with little technical skills use the Tor browser bundle. With so many ways to deanonymize the Tor browser, being aware of only one of them will not provide enough protection to the common user. More importantly, most researches available are catered towards the technical user. This article will be an easily-written compilation of the most common vulnerabilities, solutions, and advices to improve the safety of the use of The Tor Browser. This research paper caters to 1st and 2nd year CSI or IT students

INTRODUCTION

Ensuring privacy on the internet has become a hot topic in the past century, specially after the 9/11 attacks. Privacy is an essential right as human beings. It allows us to think freely without fear of judgement or repercussions. And it is an important tool to limit the government and private sector power and controls over us.

To protect our privacy, tools like VPNs, private emails and open source applications like the Tor browser has made the use of onion routing more and more common amongst the public and everyday people.

Tor is used for both licit and illicit purposes. While sometimes labeled as the dark corner of the web, referring to its past relation with Bitcoin and the virtual drug marketplace, Silk Road, it's also used by those living in authoritarian places and countries with internet censorship. News organizations also employ it to ensure the protection and privacy of whistleblowers. And on a more personal level, every day users use it to avoid targeted advertising. In our research paper, we will discuss the strengths, and more importantly the vulnerabilities and security exploits available to attack the Tor browser's ability to conceal information and user identity. We compiled a few recommendations to improve the security the Tor Bundle in the last part of the article

THREAT MODEL

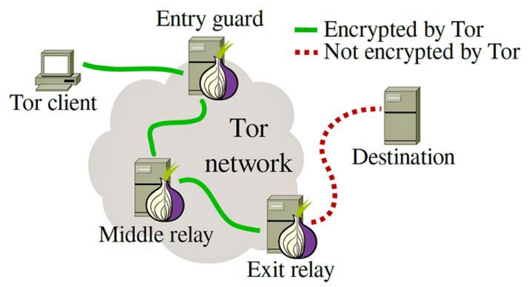
Some known threats to the anonymity of Tor are:

- Client side scripting: JavaScript and Flash code vulnerabilities can be exploited in the browser to capture users data
- Browser fingerprinting: When Tor users use a browser other than the Tor Browser Bundle over the Tor network, the browser sends information to visited sites, such as installed add-ons, version, etc. Since the number of people with matching sets of information is likely to be low, the browser fingerprint lowers anonymity, with the worst case being unique identification.
- Side channel leaks: external 3rd parties security weaknesses can compromise user privacy specially if login credentials, PII, cc numbers are sent over the network. If users use insecure networks, their information may be accessed by the exit node
- Malicious nodes: because of the open source and volunteer nature of Tor, any user can run an exit node to monitor the traffic coming off it. [1]
- Memory corruption: The Tor browser shares a large amount of code with Firefox which is mostly written in C++. C++ relies on manual memory management, trading reliability for flexibility and performance. An attacker that manages to read the memory, or leak some pointers can fingerprint a browsing session in a number of different ways.

THE TOR PROJECT

Tor is an open-source anonymity network that uses onion routing to prevent third parties from observing a user's web connection. The term onion routing refers to Tor's layered encryption (analogous to the layers in an onion), which directs Internet traffic through a worldwide, volunteer overlay network consisting of more than seven thousand relays to conceal a user's location and usage from anyone conducting network surveillance or traffic analysis.

Onion routing is implemented by encryption in the application layer of a communication protocol stack, nested like the layers of an onion. Tor encrypts the data, including the next node destination IP address, multiple times and sends it through a virtual circuit comprising successive, random-selection Tor relays. Each relay decrypts a layer of encryption to reveal the next relay in the circuit to pass the remaining encrypted data on to it. The final relay decrypts the innermost layer of encryption and sends the original data to its destination without revealing or knowing the source IP address. Because the routing of the communication was partly concealed at every hop in the Tor circuit, this method eliminates any single point at which the communicating peers can be determined through network surveillance that relies upon knowing its source and destination. [2]



MOST COMMON EXPLOITATIONS

Like all current low-latency anonymity networks, Tor cannot and does not attempt to protect against monitoring of traffic at the boundaries of the network. Tor is also subject to browser based vulnerabilities, human errors and back-end related bugs.

JavaScript vulnerabilities

JavaScript libraries can be exploited, and that can lead to scenarios that makes it possible to trace internet activities and the compromise of privacy. Embedded JavaScript code can track browser cookies. Companies can store all kinds of user-specific information in browser cookies: browser type, preferences, location, etc. Many websites are already doing it to be able to offer a better user experience.

Another common exploitation of JavaScript is through cross-site scripting (XSS). XSS is a vulnerability that allows hackers to embed malicious JavaScript code into an legitimate website, which is ultimately executed in the browser of a user who visits the website.

As of March 2020, the Tor browser fixed a bug that could have allowed JavaScript code to execute on websites even if users thought they disabled it for maximum security. The release updated Tor to 0.4.2.7 and NoScript to 11.0.19.

NoScript is an open source firefox extension that allows the execution of JavaScript, Java, Flash and other plugins on whitelisted websites. NoScript also provides anti-XSS and anti-clickjacking protections.

With the new version of Tor, JavaScript is completely "off" with no ambiguous in-between states when configured on the "safest" security setting. Tor's 'standard' setting used to enable JavaScript by default, which users could upgrade to either 'safer', which disabled JavaScript on non-HTTPS sites, or 'safest', which disabled JavaScript completely.

Pros and Cons of enabling and disabling JavaScript

The act of enabling JavaScript only on some websites could itself become an inadvertent cookie used to fingerprint users as they come up around the web. Add-ons can also introduce 3rd-party vulnerabilities the Tor browser may not have control over. The act of whitelisting some websites is handled with the NoScript extension, which have had problems on several occasions in the past.

For example, NoScript was disabled last year over a digital signature issue with Firefox. NoScript hadn't changed, its digital signature was still valid and unexpired but Firefox no longer trusted it, and so Tor Browser wouldn't (indeed, for most users, couldn't) load it any more. The bug was somewhere in Mozilla's signature verification, not in the addon itself – and the bug seemed to affect the validation of every addon of almost every version of Firefox.

In an experiment using a simple Python Script with Selenium to open the most visited websites in the US, we observed the following behaviour:

- Green: fully functional
- Yellow: limited functionality
- Red: blank page

Websites/ Security configuration	Standard	Safer	Safest
Instagram	🟩	🟩	🟥
Youtube	🟩	🟩	🟥
Facebook	🟩	🟩	🟨
Amazon	🟩	🟩	🟨
Reddit	🟩	🟩	🟨
Wikipedia	🟩	🟩	🟨
Netflix	🟩	🟩	🟨
Zoom	🟩	🟩	🟨
Blogspot	🟩	🟩	🟨
Twitter	🟩	🟩	🟨
Microsoft	🟩	🟩	🟨

Tor Browser Bundle Inefficiencies

Because of the multiple hops the network have to go through, Tor is often slow to launch and navigate. Many users have noticed a long delay between clicking "Start Tor Browser Bundle" and the Tor Browser Bundle opening. A typical scenario would be that the user would click on "Start Tor Browser Bundle". At this point, Vidalia (the graphical controller for Tor, whose interface confused users) appeared. Many users incorrectly assumed after 30 seconds or so that all their internet traffic was anonymized and proceeded to open Firefox or Internet Explorer. [3] On top of that, many have reported that browsing speed is slower than usual over high speed internet bandwidth. Some have also had trouble discriminating which window was the Tor Browser Bundle and which was a normal browser window. This caused users to (erroneously) use a non-protected Firefox session to perform study tasks. And this introduces vulnerabilities to the network that Tor is supposed to protect against, like browser fingerprinting and traffic analysis.

According to a research by Steven Murdoch and George Danezis, attackers can use traffic analysis techniques to gain only a partial view of the network to infer which nodes are being used to relay the anonymous streams. This has shown that otherwise unrelated streams can be linked back to the same initiator. This attack, however, fails to reveal the identity of the original user. [4]

Exit nodes attacks

Tor exit relays are operated by volunteers and together push more than 1 GiB/s of network traffic. By design, these volunteers are able to act as man-in-the-middle between the users and the destination, which allows them to inspect and modify the anonymized network traffic. The Tor circuits are composed of encrypted tunnels that ends at exit relays, and from there, the network traffic travels to the open internet and to the final destination. This allows exit nodes operators to perform attacks like traffic sniffing, DNS poisoning, and SSL-based attacks such as HTTPS MitM and sslstrip. [5]

Many websites from the open internet and even government entities have also implemented policies that block network traffic coming from known exit relays. For example, BBC blocks IP addresses coming from Tor nodes. The list of Tor nodes are publicly available, and this has allowed oppressive regimes to blacklist them in an attempt to control their citizens ability to stay anonymous.

Jansen et al., described a DDoS attack targeted at the Tor node software, as well as defenses against that attack and its variants. The attack works using a colluding client and server, and filling the queues of the exit node until the node runs out of memory, and hence can serve no other (genuine) clients. By attacking a significant proportion of the exit nodes this way, an attacker can degrade the network and increase the chance of targets using nodes controlled by the attacker. [6]

Memory corruption attacks

The most common way to de-anonymize Tor users is to exploit security vulnerabilities in the software used to access the Tor network. Tor is usually accessed via the Tor Browser (TB), which includes a pre-configured Tor client. Since TB is based on Mozilla's Firefox browser, they share a large part of their attack surfaces. The Firefox browser is mostly written in C and C++, which unlike modern programming languages, they rely on manual memory management, trading reliability for flexibility and performance. Hence, memory management errors often create vulnerabilities that can be exploited to hijack control flow and perform other malicious operations that were never intended by the program authors.

Traditionally, attackers used a buffer overflow to directly inject malicious code into a program and execute it. However, the introduction of the W⊕X policy that requires memory pages to either be writable or executable, but not both, made most code-injection attacks obsolete. As W⊕X became commonplace, attackers changed their tactics from code injection to code reuse. These attacks reuse existing, legitimate code for malicious purposes and have therefore proven far harder to stop than code injection. [7]

RECOMMENDATIONS

Our research is a comprehensive guide encompassing the easiest steps users with little technical knowledge can take to protect their anonymity when using the Tor browser. We've compiled materials from several authors who have delved in depth about JavaScript vulnerabilities and XSS techniques, traffic sniffing and DNS poisoning hacks, heap-based and stack exploits, and translated into a simple technical guide.

The findings from this article can be applied to improve the network in a few ways, like redesigning the user interface and experience by targeting the operational and optimization aspect of the Tor Browser Bundle.

JavaScript configuration

Gallager, Patil and Memon suggest more studies on operational and architectural modifications that reduces the attack opportunities that arises from enabling JavaScript within the Tor Browser Bundle. Additionally, they recommend to have a warning letting users know the browser default low security level at a prominent position when opening the application. [1]

Currently, as of April 2020, Tor Browser version 9.0.8 presents users with three options:

- Standard where all JavaScript function are enabled
- Safer where only some functions are disabled
- Safest where JavaScript is disabled by default.

We recommend that the Tor browser should have JavaScript set in Safer mode by default when the user initially installs the bundle. There should be an initiation page advising which configuration is more appropriate for each type of user. A dissident living in a repressive regime has different security needs than the common user from a liberal country who just wishes to remain anonymous and avoid targeted advertisement. Tor should recommend the safest option for the 1st and safer for the 2nd.

Tor latency

The Tor browser is notoriously slower than commercial browsers like Firefox, Internet Explorer or Google Chrome. Roger Dingledine and Steven J. Murdoch researched a few causes in "Why Tor is slow and what we're going to do about it:

- Tor inefficient traffic congestion control: streams that require less capacity like web-browsing are not distributed appropriately compared to more demanding ones like bulk transfers or downloading

- Limited capacity of Tor network: the current number of relays available is not enough to support the amount of users. Moreover, some relays are overloaded while others are underutilized.
- Users side network problems: some low-bandwidth users spend too much of their network overhead downloading directory information. [8] Some may not be aware they can switch the network route to a more efficient one.

Most internet users have used Tor but very few use it for all web traffic according to a poll conducted at the Privacy Enhancing Technologies Symposium in 2007. The latency is sufficiently high so as to discourage its everyday adoption. [9]

Some have suggested modifications on the data path between relays. UDP over TLS (DTLS) and IPSec have been proposed as new link protocols, although more research is required. For the common user with little technical knowledge, better user training will suffice as alternative solution. By letting users know that Tor isn't designed for data sharing and suggesting alternatives like VPN services, this should reduce the system overload.

Tor network capacity should also be expanded by adding more relays. Roger Dingledine and Steven J. Murdoch proposed encouraging more volunteers to run Tor servers, and existing volunteers to keep their servers running. They've found that going to conferences and speaking directly to attendees did help. Direct contact yielded better response than online advocacy as there are thousand as of people out there with spare fast network connections and willingness to help.

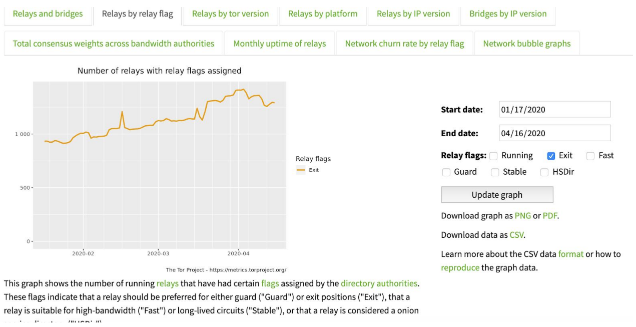
We suggest offering monetary compensations. Even a small one would probably encourage more users to take part on the Tor project. Tor should also distribute a guide with easy to follow steps and automated scripts to help users setup and maintain a relay.

Some relays advertise more bandwidth than they can realistically handle, causing higher latency in the network. These anomalies might be due to bad load balancing on the part of the Tor designers, bad rate limiting or unstable network connection on the part of the relay operator, or malicious intent.

As of April 2019, the Tor project published an article in their blog describing the use of tools like 'Torflow' and 'Simple Bandwidth Scanner' to monitor the network performance. They encourage improvements by opening tickets, developing a compatible external application programming interface, or extending the existing bandwidth file format. [10]

Malicious nodes

As of April 2020, there are between 1250-1500 exit relays distributed all around the globe serving as part of the Tor anonymity network. To prevent the most popular type of exit-node attacks, the Tor browser uses the HTTPS-Everywhere and NoScript extension from Firefox. The first translates HTTP traffic to HTTPS and the second prevents many scripting attacks. However, these don't avert attacks that may arise from web sites implementing poor security such as the lack of site-wide TLS, session cookies being sent in the clear, or servers using weak cipher suites configurations.



Start date: 01/17/2020

End date: 04/16/2020

Relay flags: □ Running ☑ Exit □ Fast □ Guard □ Stable □ HSDir

Download graph as PNG or PDF.

Download data as CSV.

Learn more about the CSV data format or how to reproduce the graph data.

This graph shows the number of running relays that have had certain flags assigned by the directory authorities. These flags indicate that a relay should be preferred for either guard ("Guard") or exit positions ("Exit"), that a relay is suitable for high-bandwidth ("Fast") or long-lived circuits ("Stable"), or that a relay is considered a onion

The Tor project also has a reporting system that allows users to flag suspicious relays. Although most of them are misconfigured. However, this doesn't mean they're taken out of the network. After getting assigned with a "BadExit" flag, these nodes are still used as middle or first hops. The flag only prevents it to be selected as exit node.

In a research by Philip Winter et al. [11] The authors suggested the implementation of Python based relays scanner exitMap and HoneyConnector to detect sniffing attacks. However, they also admitted that for both their frameworks, performing attribution was problematic. It was difficult to distinguish if the attacker was the relay operator or any other entity along the path from the exit relay to the destination. On top of that, their scanners could also be used for unethical purposes like website scraping or online voting manipulation.

Gallagher, Patil, Memon suggested displaying the IP address, country and encryption status of the selected route on the network in another research. For the common user, displaying a message at the beginning of the session warning about the dangers of improper encryption and configuration will suffice.

Additionally, the following recommendations should be presented:

- End to end encryption makes Tor more secure. Most current websites are already using HTTPS to secure communications, rather than the old, insecure HTTP standard. HTTPS is the default setting in Tor, for sites that support it. It should be noted that .onion sites don't use HTTPS as standard because communication within the Tor network, using Tor hidden services are by its very nature, encrypted.
- Another way to improve Tor safety is to use websites and services that don't report on activities as a matter of course. Switching from Google search to

DuckDuckGo reduces trackable data footprint. Switching to encrypted messaging services such as Ricochet (which can routed over the Tor network) can also improve anonymity.

- Avoid using Personal Identifying Information, logins, subscriptions and payments
- A Virtual Private Network (VPN) can protect the user from malicious exit nodes by continuing to encrypt data once it leaves the Tor network. If data remains encrypted, a malicious exit node will not have a chance to intercept it. Using Tor over a VPN like ProtonVPN can add an additional layer of protection.

REFERENCES

[1] Kevin Gallagher, Sameer Patil, Nasir Memon. New Me: Understanding Expert and Non-Expert Perceptions and Usage of the Tor Anonymity Network. July 12–14, 2017 • Santa Clara, CA, USA
ISBN 978-1-931971-39-3

[2] Termanini, Rocky (2017). The Nano Age of Digital Immunity Infrastructure Fundamentals and Applications: The Intelligent Cyber Shield for Smart Cities. CRC Press. pp. 210–211. ISBN 978-1-351-68287-9. LCCN 2017053798

[3] Greg Norcie, Kelly Caine, L Jean Camp. Eliminating Stop-Points in the Installation and Use of Anonymity Systems: a Usability Evaluation of the Tor Browser Bundle. (PDF) July 2012

[4] Murdoch, Steven J.; Danezis, George (19 January 2006). "Low-Cost Traffic Analysis of Tor" (PDF). Retrieved 21 May 2007.

[5] Philipp Winter1, Richard Köwer3, Martin Mulazzani2, Markus Huber2, Sebastian Schrittwieser2, Stefan Lindskog1, and Edgar Weippl2. Spoiled Onions: Exposing Malicious Tor Exit Relays

[6] Jansen, Rob; Tschorsch, Florian; Johnson, Aaron; Scheuermann, Björn (2014). The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network (PDF). 21st Annual Network & Distributed System Security Symposium. Retrieved 28 April 2014

[7] Mauro Conti, Stephen Crane, Tommaso Frassetto, Andrei Homescu, Georg Koppen, Per Larsen,
Christopher Liebchen, Mike Perry, and Ahmad-Reza Sadeghi. Selfrando: Securing the Tor Browser against de-anonymization Exploits. Proceedings on Privacy Enhancing Technologies ; 2016 (4):454–469

[8] Roger Dingledine, Steven J. Murdoch. Why Tor is slow and what we’re going to do about it. March 11, 2009

[9] Joel Reardon (2008). Improving Tor using a TCP-over-DTLS Tunnel. UWSpace. http://hdl.handle.net/10012/4011

[10] How Bandwidth Scanners Monitor The Tor Network. Juga.
https://blog.torproject.org/how-bandwidth-scanners-monitor-tor-network. April 11, 2019

[11] Philipp Winter, Richard Köwer, Martin Mulazzani, Markus Huber, Sebastian Schrittwieser, Stefan Lindskog, and Edgar Weippl. Spoiled Onions: Exposing Malicious Tor Exit Relays.